Fig.1
(Prior Art)

ANALOG FORM, 203

201

Fig.2

WM
WITH
SEMANTIC
DIGEST,
307

SEMANTIC INFORMATION, 305

ANALOG FORM, 303

301

Fig.3

401

Fig.4

5/11

207

205

503

ANALOG FORM

505 — SEMANTIC DIGEST READER

SEMANTICS READER — 507

OLD SEMANTIC DIGEST, 515

SEMANTICS DATA, 509

SEMANTIC DIGEST MAKER — 511

NEW SEMANTIC DIGEST, 513

COMPARATOR — 517

COMPARISON RESULT, 519
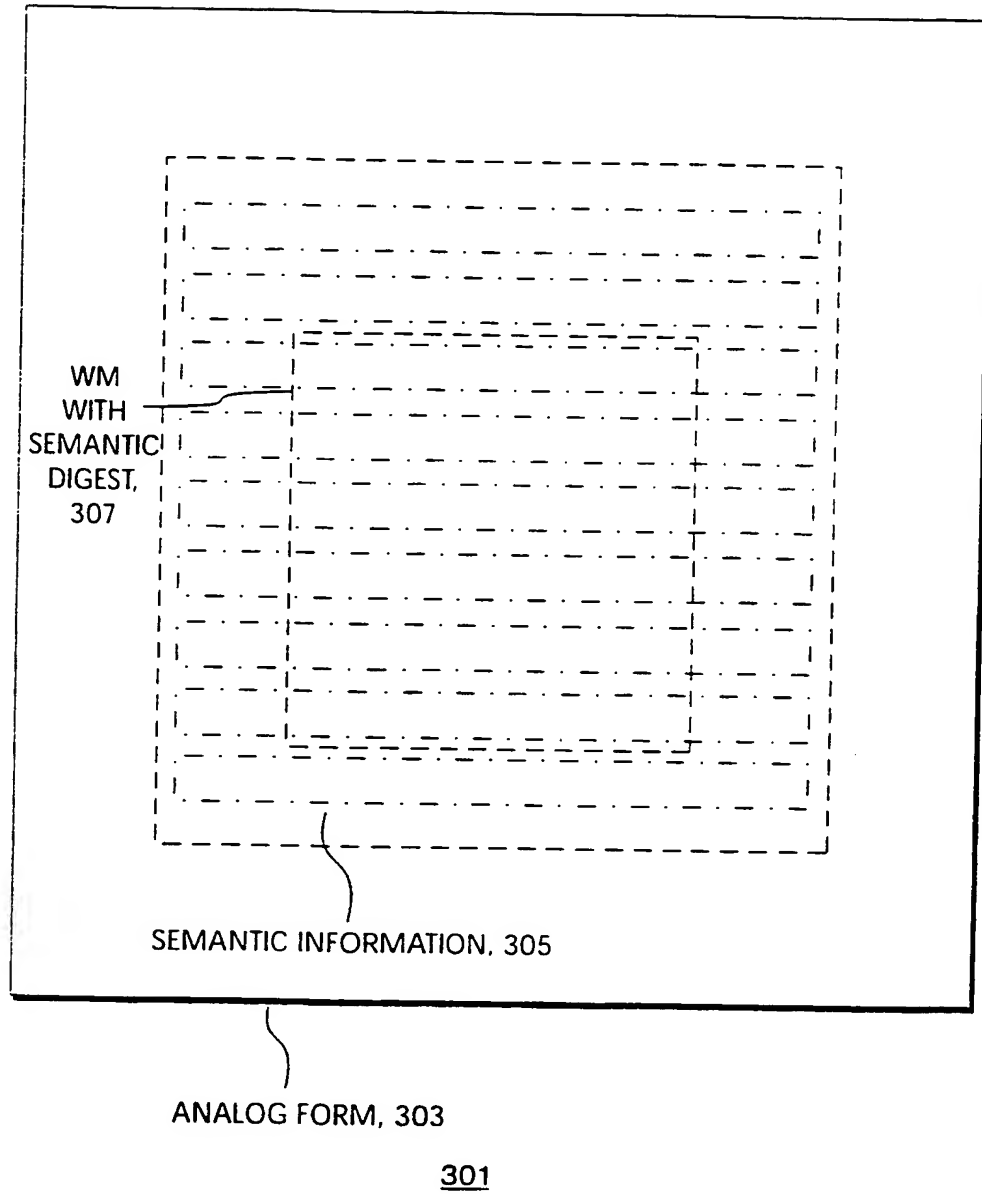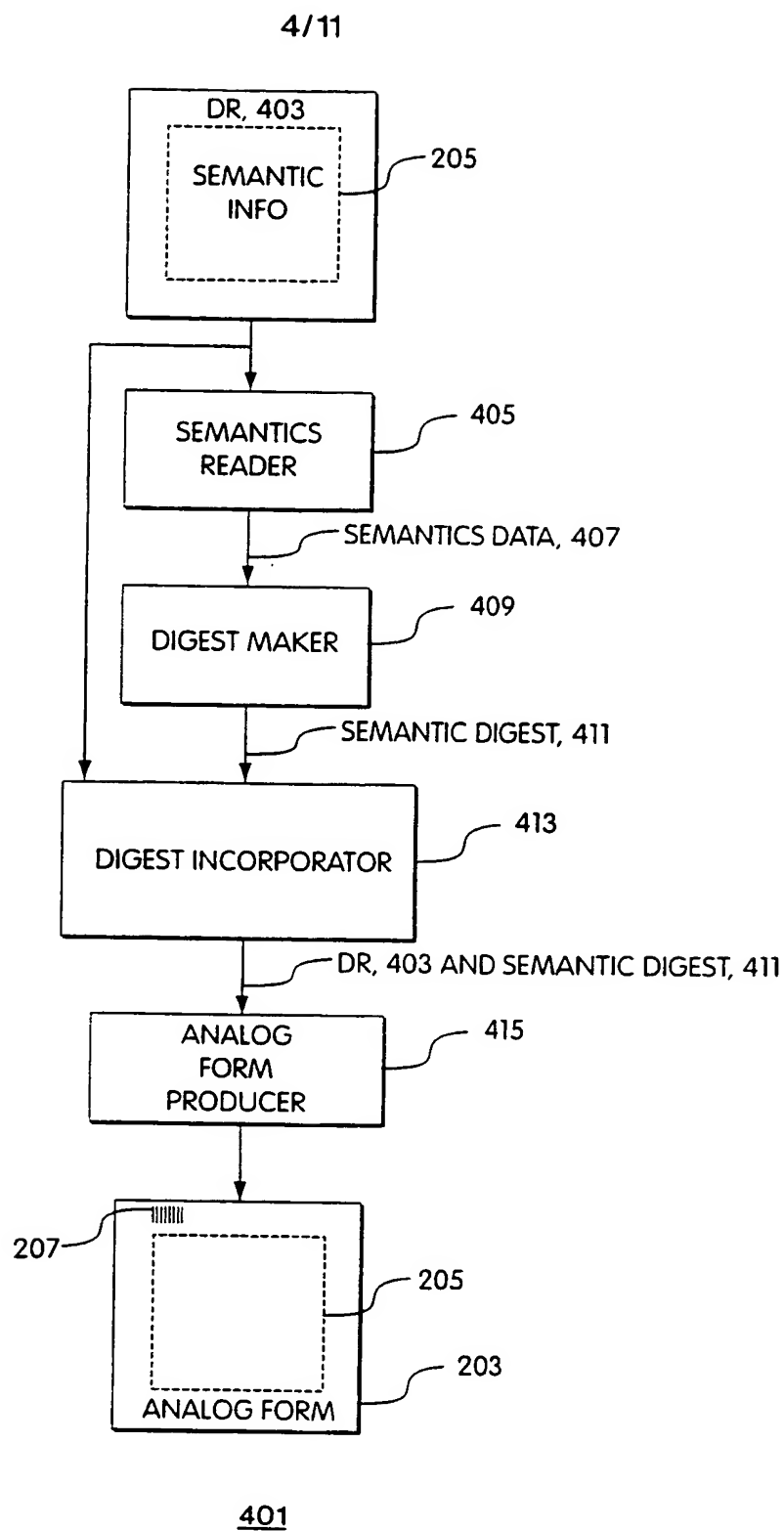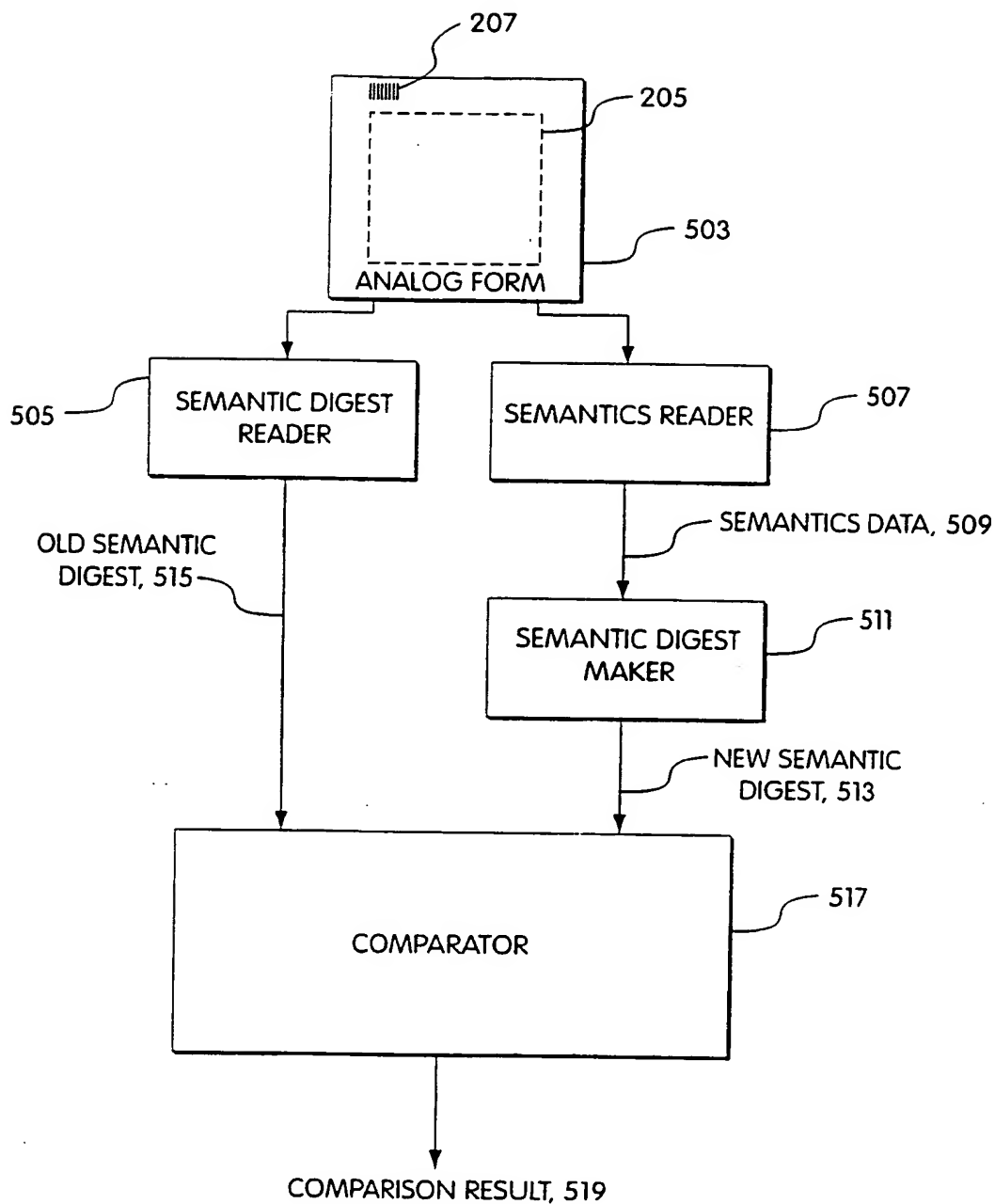
501

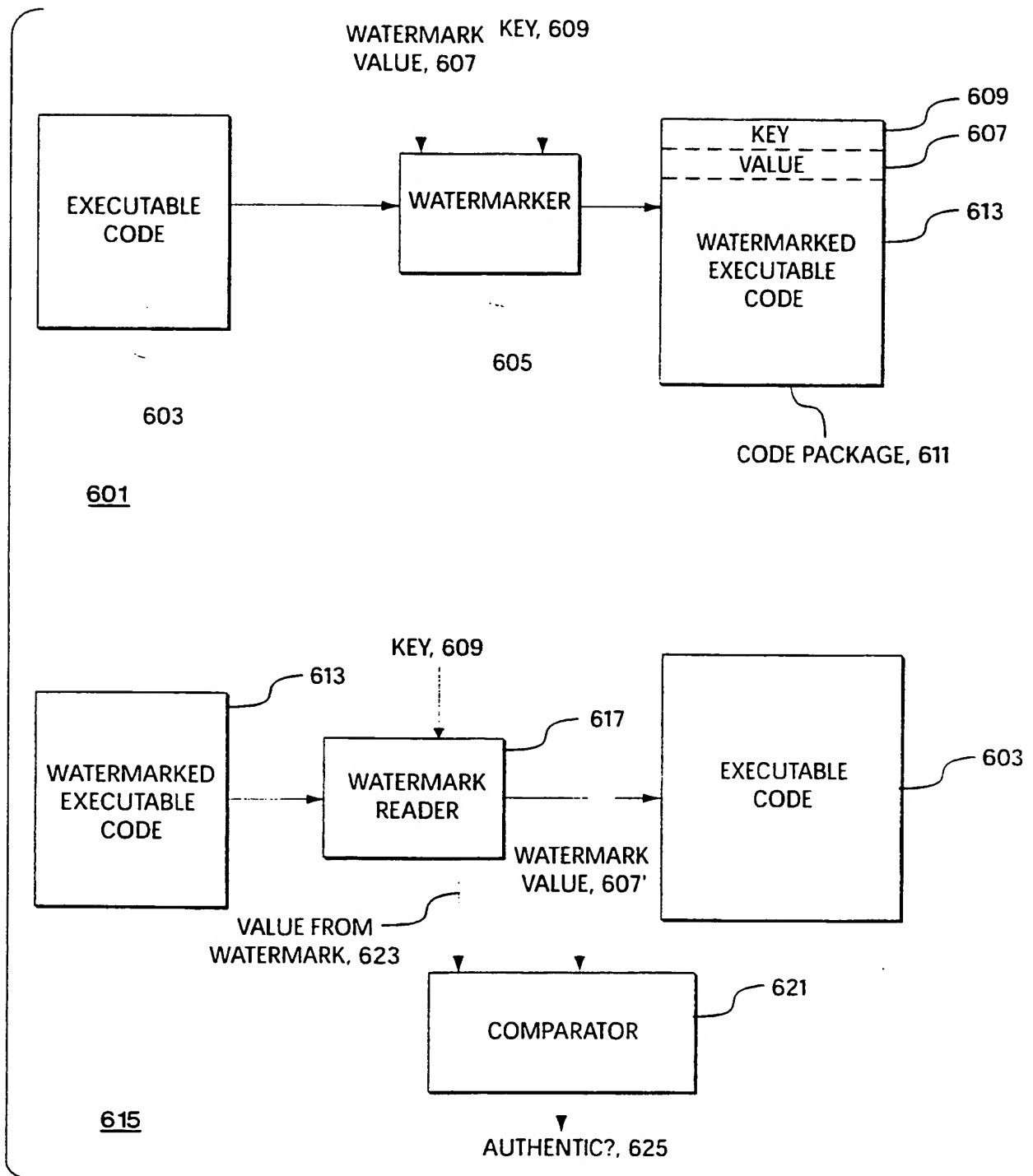Fig.5

Fig.6

7/11

```
private void add()
0:    iconst_0
1:    istore_2
2:    iconst_1
3:    istore_1
4:    goto            #14
7:    ioad_2
8:    iload_1
9:    iadd
10:   istore_2
11:   iinc        %1       1
14:   iload_1
15:   bipush      100
17:   if_icmple   #7
20:   getstatic   java.lang.System.out
                  Ljava/io/PrintStream; (48)
23:      nop
24:   new    <java.lang.StringBuffer> (32)
27:   dup
28:   ldc    "1 + 2 + ... + 100 = " (3)
30:      nop
31:   invokespecial java.lang.StringBuffer.<init>
                  (Ljava/lang/String;)V (38)
34:   iload_2
35:   invokevirtual java.lang.StringBuffer.append
                  (I)Ljava/lang/StringBuffer; (40)
38:   ldc    "" (1)
40:      nop
41:   invokevirtual java.lang.StringBuffer.append
        (Ljava/lang/String;)Ljava/lang/StringBuffer;      (41)
44:   invokevirtual java.lang.StringBuffer.toString
                  ()Ljava/lang/String; (52)
47:   invokevirtual java.io.PrintStream.println
                  (Ljava/lang/String;)V (49)
50:   return
```
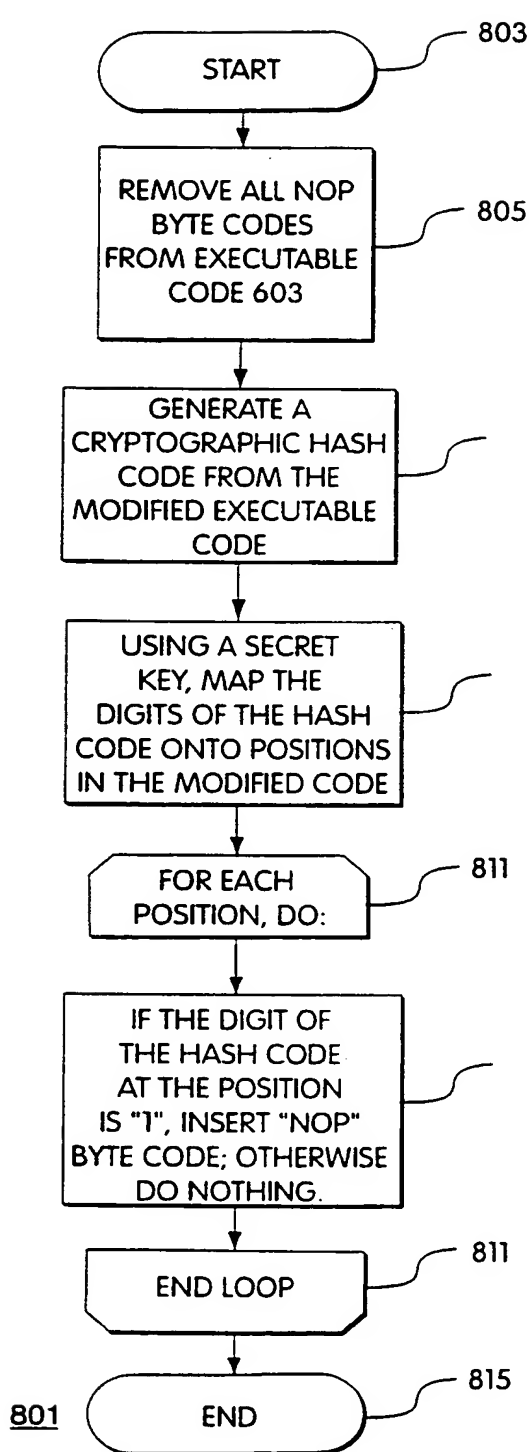
703(a) — 23

703(b) — 30

703(c) — 40

701

Fig.7

8/11

START — 803

REMOVE ALL NOP
BYTE CODES
FROM EXECUTABLE
CODE 603 — 805

GENERATE A
CRYPTOGRAPHIC HASH
CODE FROM THE
MODIFIED EXECUTABLE
CODE — 807

USING A SECRET
KEY, MAP THE
DIGITS OF THE HASH
CODE ONTO POSITIONS
IN THE MODIFIED CODE — 809

FOR EACH
POSITION, DO: — 811

IF THE DIGIT OF
THE HASH CODE
AT THE POSITION
IS "1", INSERT "NOP"
BYTE CODE; OTHERWISE
DO NOTHING. — 813

END LOOP — 811

801   END — 815

Fig.8

START — 819

USE THE SECRET
KEY TO DETERMINE
THE POSITIONS IN THE
WATERMARKED CODE
THAT CONTAIN THE
WATERMARK — 821

FOR EACH
POSITION DO — 823

IF THE BYTE CODE
AT THE POSITION IS
"NOP", THAT DIGIT OF
THE WATERMARK VALUE
IS "1"; OTHERWISE
IT IS "0". — 825

END LOOP — 823

REMOVE ALL NOP
BYTE CODES FROM
WATERMARKED
EXECUTABLE CODE
613 — 829

GENERATE A
CRYPTOGRAPHIC
HASH CODE FROM
EXECUTABLE CODE
603' — 831

HASH CODE=
WATERMARK
VALUE? — 833

NO

CODE NOT AUTHENTIC — 835

YES

CODE AUTHENTIC — 837

END — 839   817

Fig.9

```
12 putfield #250 <Field java.until.Vector nextSites>
15 aload_0
16 ldc #48 <String "Failed: ">
18 putfield #210 <Field java.lang.String failures>
21 aload_0
22 aload_1
23 putfield #291 <Field DWASuitCase suitcase>
26 aload_0
27 aload_1
28 icovst_4
29 invokevirtual #217 <Method java.lang.Object get(int)>
32 checkcast #119 <Class DWAPolicy>
35 putfield #233 <Field DWAPolicy policy>
38 aload_0
39 aload_1
40 iconst_5
41 invokevirtual #217 <Method java.lang.Object get(int)>
44 checkcast #124 <Class DWASiteInfo>
47 putfield #286 <Field DWASiteInfo siteInfo>
```
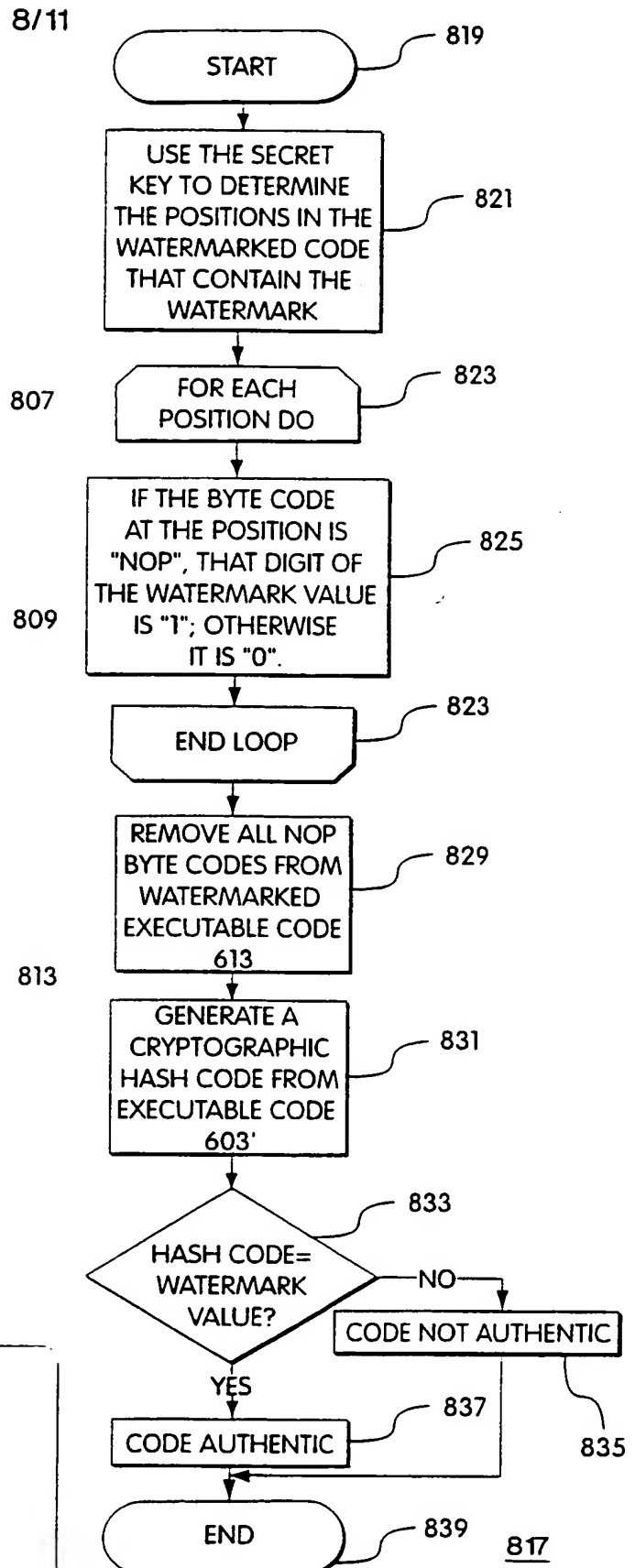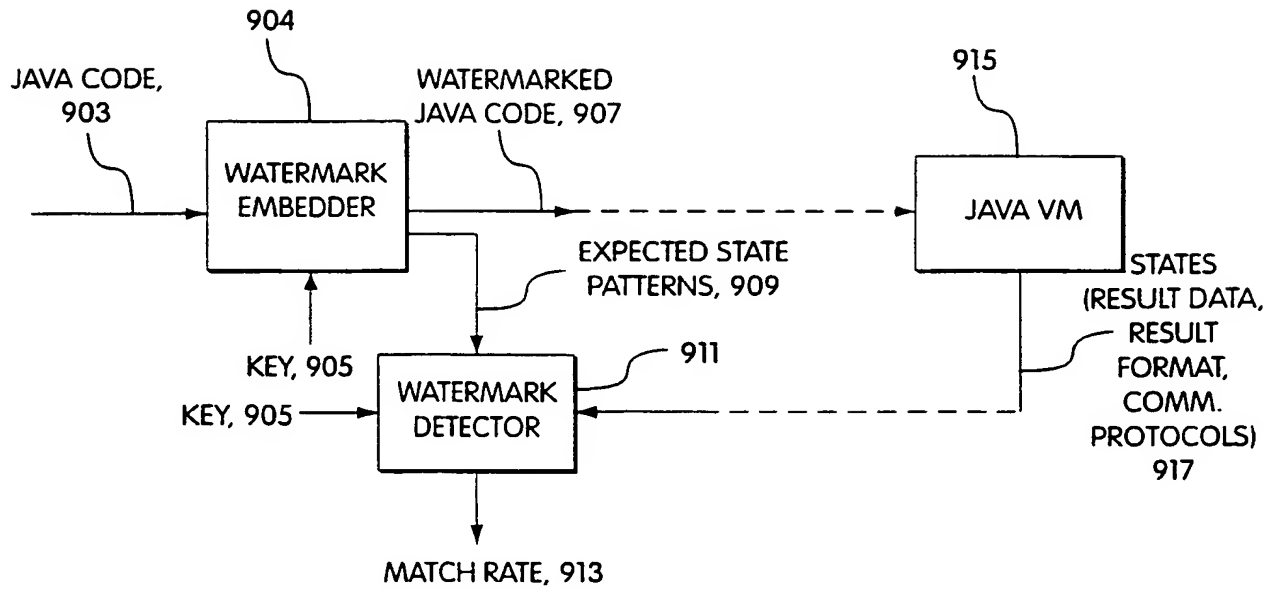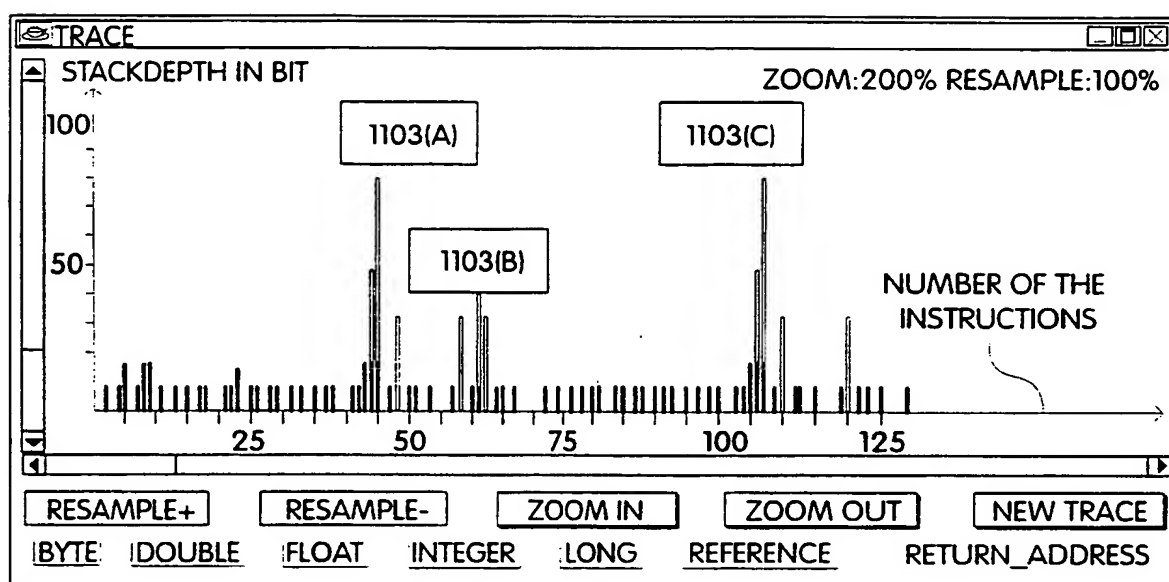
1001

Fig.10

1101   Fig. 11